UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/873,038	06/02/2001	Sanjay Lal	004906.P080	4240

8791 7590 12/04/2003

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD, SEVENTH FLOOR
LOS ANGELES, CA 90025

EXAMINER	
PATEL, HARESH N	

ART UNIT	PAPER NUMBER
2126	7

DATE MAILED: 12/04/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/873,038	LAL, SANJAY	
	Examiner	Art Unit	
	Haresh Patel	2126	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-31 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-31 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 02 June 2001 is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) ☐ The translation of the foreign language provisional application has been received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 5 and 6 .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. Claims 1-31 are presented for examination.

Specification

2. The disclosure is objected. Some of the informalities are:

i. The section "CROSS-REFERENCE TO RELATED APPLICATIONS" is missing co-pending applications and related arts.

ii. The "DETAILED DESCRIPTION OF THE INVENTION" section contains significant amount of prior art contents. All known prior art contents from the "DETAILED DESCRIPTION OF THE INVENTION" section needs to be moved into the "Description of Related Art" sub-section of the "BACKGROUND OF THE INVENTION" section.

Appropriate correction is required.

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

The following title is suggested: "Method and apparatus to handle exceptions using a dedicated exception handling processor in a multiprocessor environment ".

Drawings

4. Figure 1 should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g). A proposed drawing correction or corrected drawings

are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Information Disclosure Statement

5. An initialed and dated copy of Applicant's IDS form 1449, Paper No. 5 and 6, is attached to the instant Office action.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kranich et. al. (Hereafter Kranich) in view of Yoshioka et. al. 6,425,039 (Hereafter Yoshioka).

8. As per claims 1, 12, and 21, Kranchi teaches the following:

a method for handling a number of exceptions within a processor in a multiprocessing system, the method comprising,

a system comprising,

a machine-readable medium that provides instructions for handling a number of exceptions within a processor in a multi-processing system, which when executed by a machine, causes the machine to perform operations comprising:

receiving an exception within the processor (e.g., the mechanism allows the handling of exceptions and interruptions in a multithreaded multiprocessor computer, while hiding the multiprocessor nature of the computer from the operating system, abstract), wherein each processor in the multi-processor system shares a same memory (e.g., main memory connected to the processors, figure 15); wherein the number of instructions cause the processor to determine an identification of the processor based on a query that is internal to the processor (e.g., FIG. 12 is a block diagram showing one embodiment of a mapping table. Included in FIG. 12 are thread control device 300 and mapping table unit 350. Mapping table unit 350 may be within thread control device 300 or may be a completely separate unit. In the embodiment of FIG. 12, mapping table unit 350 includes four entries 1310, each of which may contain a physical processor number. In general, the mapping table may have a number of entries equal to the number of processors in the multiprocessor computer. When indexed with a logical processor number 1302, mapping table unit 350 conveys the entry 1304, which corresponds to the index 1302. Logical processor number 1302 corresponds to the processor number contained within a thread instruction (e.g., RdFrProc2, yy, xx), col. 19, line 34 – col. 20, line 21), and modifying execution flow of the exception to execute an interrupt handler located within one of a number of different interrupt handling vector address spaces (e.g., Exceptions and interrupts may be identified by the processor by a number called a vector. This vector may then be used to index into a table which contains the address of a handling routine called a "handler". FIG. 10 shows a diagram of how a vector may be used to locate the appropriate handler. FIG. 10 includes vector 1110, vector table 1102 and memory segment 1104. Vector table includes addresses for 12 handlers, 0-11. Memory segment 1104 includes handler code 1106. Vector table 1102 may be

located in main memory. Upon receiving an exception or interrupt, the processor utilizes the associated vector 1110 to index into vector table 1102. The entry 1118 in table 1102 corresponding to vector 1110 contains the address of the handler for the interrupt or exception. Arrow 1120 indicates the beginning address of handler code 1106 in memory segment 1104. Once identified, the processor executes handler code 1106 and returns to the previously interrupted process, col. 19, line 34 – col. 20, line 21).

However, Kranich does not specifically teach a concept of a common interrupt handling vector.

Yoshioka teaches the following:

executing a number of instructions at an address within a common interrupt handling vector address space of the same memory (e.g., concept of handling common exception events, figure 2 – figure 15).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Kranich with the teachings of Yoshioka in order to handle an exception in a multiprocessor environment using the common exception handling vector to identify the processor and to further handle the exception based on the instructions specific to the occurred exception.

9. As per claims 2-6, 13-16, 22-26, Kranchi teaches the following:

each processor in the multi-processor system executes one of a number of operating systems (e.g., the multiprocessor nature of the computer to be hidden from the operating system, while minimizing the overhead necessary for proper handling, abstract),

each of the number of operating systems is associated with one of the number of different interrupt handling vector address spaces (e.g., handler code for individual vector address spaces based on operating system, figure 10),

the query that is internal to the processor includes reading a bit within a register that internal to the processor, the register is not dedicated to determining the identification of the processor (e.g., Processor 10 supports out of order execution, and thus employs reorder buffer 32 to keep track of the original program sequence for register read and write operations, to implement register renaming, to allow for speculative instruction execution and branch misprediction recovery, and to facilitate precise exceptions. A temporary storage location within reorder buffer 32 is reserved upon decode of an instruction that involves the update of a register to thereby store speculative register states, col. 4, line 64 – col. 7, line 38),

determining the identification of the processor during initialization of the processor, based communications with a memory controller that is coupled between the processors in the multiple processor system and the same memory (e.g. FIG. 12 is a block diagram showing one embodiment of a mapping table. Included in FIG. 12 are thread control device 300 and mapping table unit 350. Mapping table unit 350 may be within thread control device 300 or may be a completely separate unit. In the embodiment of FIG. 12, mapping table unit 350 includes four entries 1310, each of which may contain a physical processor number. In general, the mapping table may have a number of entries equal to the number of processors in the multiprocessor computer. When indexed with a logical processor number 1302, mapping table unit 350 conveys the entry 1304 which corresponds to the index 1302. Logical processor number 1302

corresponds to the processor number contained within a thread instruction (e.g., RdFrProc2, yy, xx), col. 19, line 34 – col. 20, line 21).

10. As per claims 7 and 27, Kranchi teaches the following:

handling of the handler code based on the vector location (e.g., vector versus handler code, figure 10).

However, Kranich does not specifically teach a concept of a common interrupt handling vector.

Yoshioka teaches the following:

instructions at an address within the common interrupt handling vector address space of the same memory (e.g., concept of handling common exception events, figure 2 – figure 15).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Kranich with the teachings of Yoshioka in order to handle an exception in a multiprocessor environment using the common exception handling vector to identify the processor and to further handle the exception based on the instructions specific to the occurred exception.

11. As per claims 8, 17, and 28, Kranchi teaches the following:

a method comprising,

a system comprising,

a machine-readable medium that provides instructions, which when executed by a machine, causes the machine to perform operations comprising:

receiving an exception within a processor wherein the processor is included in a multi-processor system (e.g., A mechanism for exception and interrupt handling in multithreaded multiprocessors is provided. The mechanism allows the handling of exceptions and interruptions in a multithreaded multiprocessor computer, while hiding the multiprocessor nature of the computer from the operating system, abstract), wherein each processor in the multi-processor system executes one of a number of operating systems (e.g., A mechanism for exception and interrupt handling in multithreaded multiprocessors is provided. The mechanism allows the handling of exceptions and interruptions in a multithreaded multiprocessor computer, while hiding the multiprocessor nature of the computer from the operating system, abstract), determining the type of exception received within the processor; (e.g. FIG. 12 is a block diagram showing one embodiment of a mapping table. Included in FIG. 12 are thread control device 300 and mapping table unit 350. Mapping table unit 350 may be within thread control device 300 or may be a completely separate unit. In the embodiment of FIG. 12, mapping table unit 350 includes four entries 1310, each of which may contain a physical processor number. In general, the mapping table may have a number of entries equal to the number of processors in the multiprocessor computer. When indexed with a logical processor number 1302, mapping table unit 350 conveys the entry 1304 which corresponds to the index 1302. Logical processor number 1302 corresponds to the processor number contained within a thread instruction (e.g., RdFrProc2, yy, xx), col. 19, line 34 – col. 20, line 21) wherein each processor in the multiprocessor system shares a same memory wherein the number of instructions cause the processor to read a bit within an internal register to determine an identification of the processor in the multi-processor system; and modifying execution flow of the exception to execute an

interrupt handler located within one of the number of different interrupt handling address spaces (e.g., Processor 10 supports out of order execution, and thus employs reorder buffer 32 to keep track of the original program sequence for register read and write operations, to implement register renaming, to allow for speculative instruction execution and branch misprediction recovery, and to facilitate precise exceptions. A temporary storage location within reorder buffer 32 is reserved upon decode of an instruction that involves the update of a register to thereby store speculative register states, col. 4, line 64 – col. 7, line 38).

However, Kranich does not specifically teach a concept of a common interrupt handling vector.

Yoshioka teaches the following:

the same memory includes a common interrupt handling address space and a number of different interrupt handling address spaces associated with each of the different processors in the multi-processor system; executing a number of instructions at an address within the common interrupt handling address space of the same memory (e.g., concept of handling common exception events, figures 2 – 15, col. 1, line 7 – col. 25, line 67).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Kranich with the teachings of Yoshioka in order to handle an exception in a multiprocessor environment using the common exception handling vector to identify the processor and to further handle the exception based on the instructions specific to the occurred exception.

12. Claims 1, 8, 12, 17, 21 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brenner, JR. et. al. US Patent application publication 2002/007131, June 13, 2002 (Hereafter Brenner) in view of Browning et. al. 6,006,247.

13. As per claims 1, 12, and 21, Brenner teaches the following:

a method for handling a number of exceptions within a processor in a multiprocessing system, the method comprising,

a system comprising,

a machine-readable medium that provides instructions for handling a number of exceptions within a processor in a multi-processing system, which when executed by a machine, causes the machine to perform operations comprising:

receiving an exception within the processor wherein each processor in the multi-processor system (e.g., The embodiment(s) of the present invention may be implemented as part of a Board Support Package (BSP) for such processors. The embodiment(s) may also be implemented as part of the kernel of an operating system for such processors, page 2, paragraph 0016) shares a same memory (e.g., A unified interrupt handling system and method is provided for an embeddable processor having multiple interrupt types. An instruction is inserted into the first vector address that disables the second interrupt mode. At the second vector address, an other instruction is inserted that branches to a common interrupt dispatcher. The common interrupt dispatcher is provided with an interrupt routine that processes the interrupt, and then re-enables the second interrupt modes. Interrupt requests are then processed by the common interrupt dispatcher without interruption, abstract, Additional steps include inserting into a first vector address of an interrupt vector table, an instruction that disables subsequent interrupts. An

other instruction is inserted at a second vector address of the exception vector table, that branches to the common interrupt dispatcher. An interrupt of the first type is received, followed by branching to the first vector address, and setting the mode identifier to indicate an interrupt of the first type was received. The instruction to disable the first and second interrupts is executed, and the other instruction is executed to branch to the common interrupt dispatcher. The interrupt is processed by the common interrupt dispatcher without interruption, and the first and second interrupts are re-enabled, e.g., page 1, paragraph 0001 – page 2, paragraph 0011),

modifying execution flow of the exception to execute an interrupt handler located within one of a number of different interrupt handling vector address spaces (e.g., dispatcher dispatching exceptions to the different handlers, figure 2),

executing a number of instructions at an address within a common interrupt handling vector address space of the same memory (e.g., concept of handling common exception events, figure 2, The processor also includes a mode status indicator to indicate the current mode of the processor, and an interrupt vector table having a first vector address executable upon receipt of an interrupt request of the first type, and a second vector address executable upon receipt of an interrupt request of the second type, the first vector address preceding the second vector address in the vector table. The processor is adapted to execute at least one instruction in the interrupt vector table without interruption, upon receipt of an interrupt request, page 2, paragraph 0010).

However, Brenner does not specifically teach a concept of determining an identification of the processor.

Yoshioka teaches the following:

the number of instructions cause the processor to determine an identification of the processor based on a query that is internal to the processor (e.g., processor list containing identified processors, figure 2).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Brenner with the teachings of Yoshioka in order to handle an exception in a multiprocessor environment using the common exception handling vector to identify the processor and to further handle the exception based on the instructions specific to the occurred exception.

14. As per claims 8, 17, and 28, Brenner teaches the following:

a method comprising,

a system comprising,

a machine-readable medium that provides instructions, which when executed by a machine, causes the machine to perform operations comprising:

receiving an exception within a processor wherein the processor is included in a multi-processor system wherein each processor in the multi-processor system executes one of a number of operating systems (e.g., The embodiment(s) of the present invention may be implemented as part of a Board Support Package (BSP) for such processors. The embodiment(s) may also be implemented as part of the kernel of an operating system for such processors, page 2, paragraph 0016),

determining the type of exception received within the processor; wherein each processor in the multiprocessor system shares a same memory wherein the number of instructions cause the

processor to read a bit within an internal register (e.g., A unified interrupt handling system and method is provided for an embeddable processor having multiple interrupt types. An instruction is inserted into the first vector address that disables the second interrupt mode. At the second vector address, an other instruction is inserted that branches to a common interrupt dispatcher. The common interrupt dispatcher is provided with an interrupt routine that processes the interrupt, and then re-enables the second interrupt modes. Interrupt requests are then processed by the common interrupt dispatcher without interruption, abstract, Additional steps include inserting into a first vector address of an interrupt vector table, an instruction that disables subsequent interrupts. An other instruction is inserted at a second vector address of the exception vector table, that branches to the common interrupt dispatcher. An interrupt of the first type is received, followed by branching to the first vector address, and setting the mode identifier to indicate an interrupt of the first type was received. The instruction to disable the first and second interrupts is executed, and the other instruction is executed to branch to the common interrupt dispatcher. The interrupt is processed by the common interrupt dispatcher without interruption, and the first and second interrupts are re-enabled, e.g., page 1, paragraph 0001 – page 2, paragraph 0011),

and modifying execution flow of the exception to execute an interrupt handler located within one of the number of different interrupt handling address spaces (e.g., dispatcher dispatching exceptions to the different handlers, figure 2, concept of handling common exception events, figure 2, The processor also includes a mode status indicator to indicate the current mode of the processor, and an interrupt vector table having a first vector address executable upon receipt of an interrupt request of the first type, and a second vector address executable upon

receipt of an interrupt request of the second type, the first vector address preceding the second vector address in the vector table. The processor is adapted to execute at least one instruction in the interrupt vector table without interruption, upon receipt of an interrupt request, page 2, paragraph 0010).

However, Brenner does not specifically teach a concept of a common interrupt handling vector.

Yoshioka teaches the following:

determine an identification of the processor in the multi-processor system (e.g., processor list containing identified processors, figure 2).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Brenner with the teachings of Yoshioka in order to handle an exception in a multiprocessor environment using the common exception handling vector to identify the processor and to further handle the exception based on the instructions specific to the occurred exception.

Conclusion

15. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

See Form PTO-892.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Haresh Patel whose telephone number is (703) 605-5234. The examiner can normally be reached on Monday, Tuesday, Thursday and Friday from 10:00 am to 8:00 pm.